

# Blankets Joint Posterior score for learning Markov network structures

Federico Schlüter<sup>a</sup>, Yanela Strappa<sup>a</sup>, Diego Milone<sup>b</sup>, Facundo Bromberg<sup>a</sup>

<sup>a</sup>*DHARMa Lab, Dept of Information Systems. Facultad Regional Mendoza, Universidad Tecnológica Nacional, Mendoza, Argentina. Tel.: +54-261-5240066*

<sup>b</sup>*Research Institute for Signals, Systems and Computational Intelligence, sinc(i), FICH-UNL/CONICET Santa Fe, Argentina.*

---

## Abstract

Markov networks are extensively used to model complex sequential, spatial, and relational interactions in a wide range of fields. By learning the structure of independences of a domain, more accurate joint probability distributions can be obtained for inference tasks or, more directly, for interpreting the most significant relations among the variables. Recently, several researchers have investigated techniques for automatically learning the structure from data by obtaining the probabilistic maximum-a-posteriori structure given the available data. However, all the approximations proposed decompose the posterior of the whole structure into local sub-problems, by assuming that the posteriors of the Markov blankets of all the variables are mutually independent. In this work, we propose a scoring function for relaxing such assumption. The *Blankets Joint Posterior* score computes the joint posterior of structures as a joint distribution of the collection of its Markov blankets. Essentially, the whole posterior is obtained by computing the posterior of the blanket of each variable as a conditional distribution that takes into account information from other blankets in the network. We show in our experimental results that the proposed approximation can improve the sample complexity of state-of-the-art scores when learning complex networks, where the independence assumption between blanket variables is clearly incorrect.

---

\*Corresponding author

Email address: federico.schluter@frm.utn.edu.ar (Federico Schlüter)

*Keywords:* Markov network, structure learning, scoring function, blankets posterior, irregular structures

---

## 1. Introduction

A Markov network (MN) is a popular probabilistic graphical model that efficiently encodes the joint probability distribution for a set of random variables of a specific domain [1, 2, 3]. MNs usually represent probability distributions by using two interdependent components: an independence structure, and a set of numerical parameters over the structure. The first is a qualitative component that represents structural information about a problem domain in the form of conditional independence relationships between variables. The numerical parameters are a quantitative component that represents the strength of the dependences in the structure. There is a large list of applications of MNs in a wide range of fields, such as computer vision and image analysis [4, 5, 6], computational biology [7], biomedicine [8, 9], and evolutionary computation [10, 11], among many others. For some of these applications, the model can be constructed manually by human experts, but in many other problems this can become unfeasible, mainly due to the dimensionality of the problem.

Learning the model from data consists of two interdependent problems: learning the structure; and given the structure, learning its parameters. This work focuses on the task of learning the structure, which is useful for a variety of tasks. The structures learned may be used to construct accurate models for inference tasks (such as the estimation of marginal and conditional probabilities) [12, 13, 14], and may also be interesting per se, since they can be used as interpretable models that show the most significant interactions of a domain [15, 16, 17, 18, 19]. The first scenario is known in practice as the density estimation goal of learning, and the second one is known as the knowledge discovery goal of learning [Chapter 16 [3]].

An interesting approach to MN structure learning is to use constraint-based (also known as independence-based) algorithms [20, 21, 22, 23]. Such algorithms

proceed by performing statistical independence tests on data, and discard all structures inconsistent with the tests. This is an efficient approach, and it is correct under the assumption that the distribution can be represented by a graph, and that the tests are reliable. However, the algorithms that follow this approach are quite sensitive to errors in the tests, which may be unreliable for large conditioning sets [20, 3]. A second approach to MN structure learning is to use score-based algorithms [24, 25, 15, 26]. Such algorithms formulate the problem as an optimization, combining a strategy for searching through the space of possible structures with a scoring function measuring the fitness of each structure to the data. The structure learned is the one that achieves the highest score.

It is important to mention that both constraint-based and score-based approaches have been originally motivated by distinct learning goals. According to the existing literature [3], constraint-based methods are generally designed for the knowledge-discovery goal of learning [22, 21], and their quality is often measured in terms of the correctness of the structure learned (structural errors). In contrast, most score-based approaches have been designed for the density estimation goal of learning [12, 13, 14], and they are in general evaluated in terms of inference accuracy. For this reason, score-based algorithms often work by considering the whole MN at once during the search, interleaving the parameter learning step. This makes them more accurate for inference tasks. However, since learning the parameters is known to be NP-hard for MNs [27], it has a negative effect on their scalability.

Recently, there has been a surge of interest towards efficient methods based on a strategy that follows a score-based approach, but with the knowledge discovery goal in mind. Basically, an undirected graph structure is learned by obtaining the probabilistic maximum-a-posteriori structure given the available data [28, 19]. This hybrid strategy achieves scalability, as well as reliable performance. Such contributions consist in the design of efficient scoring functions for MN structures, expressing the problem formally as follows: given a complete

training data set  $D$ , find an undirected graph  $G^*$  such that

$$G^* = \arg \max_{G \in \mathcal{G}} \Pr(G|D), \quad (1)$$

where  $\Pr(G|D)$  is the posterior probability of a structure given  $D$ , and  $\mathcal{G}$  is the family of all the possible undirected graphs for the domain size. This class of algorithms has been shown to outperform constraint-based algorithms in the quality of the learned structures, with equivalent computational complexities. The method proposed in this paper follows this approach.

Since there are no feasible exact methods for computing the posterior of MN structures, different approximations have been proposed. An important assumption commonly made by the current state-of-the-art methods is to suppose that the posterior of the structure is decomposable [29, 30, 3, 28, 19]. It means that the whole posterior can be computed as a product of the posteriors of the Markov blankets that compose the structure, which are smaller posteriors that can be computed independently. In fact, this is a good approximation that improves the efficiency of search. The research line of this work aims at designing a better approximation to the posterior, by relaxing such independence assumption. For this, the contribution of this work is the *Blankets Joint Posterior* (BJP), a scoring function that poses  $\Pr(G|D)$  as the joint posterior probability of the Markov blankets of  $G$ . This is achieved by formulating  $\Pr(G|D)$  in a novel way that relaxes the independence assumption between the blankets. Essentially, the whole posterior is obtained by computing the posterior of the blanket of each variable as a conditional distribution that takes into account information from other blankets in the network. In the experiments we show that the proposed approximation can improve the sample complexity of state-of-the-art scores when learning networks with complex topologies, that commonly appear in real-world problems.

After providing some preliminaries, notations and definitions in Section 2, we introduce the BJP scoring function in Section 3. Section 4 presents the experimental results for several study cases. Finally, Section 5 summarizes this work, and poses several possible directions of future work.

## 2. Background

We begin by introducing the notation used for MNs. Then we provide some additional background about these models and the problem of learning their independence structure, and also discuss the state-of-the-art of MN structure learning.

### 2.1. Markov networks

Have  $V$  as a finite set of indexes, lowercase subscripts for denoting particular indexes, e.g.,  $i, j \in V$ , and uppercase subscripts for subsets of indexes, e.g.,  $W \subseteq V$ . Let  $X_V$  be the set of random variables of a domain, denoting single variables as single indexes in  $V$ , e.g.,  $X_i, X_j \in X_V$  when  $i, j \in V$ . For a MN representing a probability distribution  $P(X_V)$ , its two components are denoted as follows:  $G$ , and  $\theta$ .  $G$  is the structure, an undirected graph  $G = (V, E)$  where the nodes  $V = \{0, \dots, n-1\}$  are the indices of each random variable  $X_i$  of the domain, and  $E \subseteq \{V \times V\}$  is the edge set of the graph. A node  $i$  is a neighbor of  $j$  when the pair  $(i, j) \in E$ . The edges encode direct probabilistic influence between the variables. Similarly, the absence of an edge manifests that the dependence could be mediated by some other subset of variables, corresponding to conditional independences between these variables.

A variable  $X_i$  is conditionally independent of another non-adjacent variable  $X_j$  given a set of variables  $X_Z$  if  $\Pr(X_i \mid X_j, X_Z) = \Pr(X_i \mid X_Z)$ . This is denoted by  $\langle X_i \perp X_j \mid X_Z \rangle$  (or  $\langle X_i \not\perp X_j \mid X_Z \rangle$  for the dependence assertion). As proven by [31], the independences encoded by  $G$  allow the decomposition of the joint distribution into simpler lower-dimensional functions called factors, or potential functions. The distribution can be factorized as the product of the potential functions  $\phi_c(V_c)$  over each clique  $V_c$  (i.e., each completely connected sub-graph) of  $G$ , that is

$$P(V) = \frac{1}{Z} \prod_{c \in \text{cliques}(G)} \phi_c(V_c), \quad (2)$$

where  $Z$  is a constant that normalizes the product of potentials. Such potential functions are parameterized by the set of numerical parameters  $\theta$ .

For each variable  $X_i$  of a MN, its Markov blanket is composed by the set of all its neighbor nodes in the graph. Hereon we denote the blanket of a variable  $X_i$  as  $B^{X_i}$ . An important concept that is satisfied by MNs is the Local Markov property, formally described as:

**Local Markov property.** A variable is conditionally independent of all its non-neighbor variables given its MB. That is

$$\langle X_i \perp \{X_V \setminus B^{X_i}\} | B^{X_i} \rangle. \quad (3)$$

By using such property, the conditional independences of  $P(X_V)$  can be read from the structure  $G$ . This is done by considering the concept of separability. Each pair of non-adjacent variables  $(X_i, X_j)$  is said to be separated by a set of variables  $X_Z \subseteq X_V \setminus \{X_i, X_j\}$  when every path between  $X_i$  and  $X_j$  in  $G$  contains some node in  $X_Z$  [1].

In machine learning, statistical independence tests are a well-known tool to decide whether a conditional independence is supported by the data. Examples of independence tests used in practice are Mutual Information [32], Pearson's  $\chi^2$  and  $G^2$  [33], the Bayesian statistical test of independence [34], and the Partial Correlation test for continuous Gaussian data [20]. Such tests require the construction of a contingency table of counts for each complete configuration of the variables involved; as a result, they would have an exponential cost in the number of variables [35]. For this reason, the use of the local Markov property has a positive effect for learning independence structures, allowing the use of smaller tests. Accordingly, the BJP score introduced in this work takes advantage of this property by computing a set of conditional probabilities that are more reliable and less expensive.

## 2.2. MN structure learning

The MN structure is learned from a training dataset  $D = \{D_1, \dots, D_d\}$ , assumed to be a representative sample of the underlying distribution  $P(X_V)$ . Commonly,  $D$  has a tabular format, with a column for each variable of the

domain  $X_V$ , and one row per data point. This work assumes that each variable is discrete, with a finite number of possible values, and that no data point in  $D$  has missing values. As mentioned in the introduction, this work focuses on methods for computing  $\Pr(G|D)$ . For this reason, in this subsection we discuss two recently proposed scoring functions that approximate it: the Marginal Pseudo Likelihood (MPL) score [19], and the Independence-based score (IB-score) [28].

In MPL, each graph is scored by using an efficient approximation to the posterior probability of structures given the data. This score approximates the posterior by considering  $P(G | D) \propto P(D | G) \times P(G)$ . Since the data likelihood of the graph,  $P(D | G)$ , is in general extremely hard to evaluate, MPL utilizes the well-known approximation called the pseudo-likelihood [36]. This score was proved to be consistent, that is, in the limit of infinite data the solution structure has the maximum score. For finding the MPL-optimal structure, two algorithms were presented: an exact algorithm using pseudo-boolean optimization, and a fast alternative to the exact method, which uses greedy hill-climbing with near-optimal performance. This algorithm learns the blanket for each variable, locally optimizing the MPL for each node, independently of the solutions of the other nodes. For this, it uses an approximate deterministic hill-climbing procedure similar to the well-known IAMB algorithm [37]. Finally, a global graph discovery method is applied by using a greedy hill-climbing algorithm, searching for the structure with maximum MPL score, but only restricting the search space to the conflicting edges.

The independence-based score (IB-score) [28] is also based on the computation of the posterior, but using the statistics of a set of conditional independence tests. In this score the posterior  $\Pr(G | D)$  is computed by combining the outcomes of a set of conditional independence assertions that completely determine  $G$ . Such set was called the *closure* of the structure, denoted  $\mathcal{C}(G)$ . Thus, when using IB-score the problem of structure learning is posed as the maximization of the posterior of the closure for each structure. Formally,

$$G^* = \arg \max_G \Pr(\mathcal{C}(G) | D). \quad (4)$$

Applying the chain rule over the posterior of the closure,

$$\Pr(\mathcal{C}(G) \mid D) = \prod_{c_i \in \mathcal{C}(G)} \Pr(c_i \mid c_1, \dots, c_{i-1}, D), \quad (5)$$

the IB-score approximates such probability by assuming that all the independence assertions  $c_i$  in the closure  $\mathcal{C}(G)$  are mutually independent. The resulting scoring function is computed as:

$$\text{IB-score}(G) = \prod_{c_i \in \mathcal{C}(G)} \log \Pr(c_i \mid D), \quad (6)$$

where each term  $\log \Pr(c_i \mid D)$  is computed by using the Bayesian statistical test of conditional independence [34, 38]. Together with the IB-score, an efficient algorithm called IBMAP-HC is presented to learn the structure by using a heuristic local search over the space of possible structures.

### 3. Blankets Joint Posterior scoring function

We introduce now our main contribution, the Blankets Joint Posterior (BJP) scoring function. Consider some graph  $G$  representing the independence structure of a positive MN. It is a well-known fact that, by exploiting the graphical properties of such models, the independence structure can be decomposed as the unique collection of the blankets of the variables [3, Theorem 4.6 on p. 121]. Thus, the computation of the posterior probability of  $G$  given a dataset  $D$  is equivalent to the joint posterior of the collection of blankets of  $G$ , that is,

$$\Pr(G \mid D) = \Pr(B^{X_0}, B^{X_1}, \dots, B^{X_{n-1}} \mid D). \quad (7)$$

In contrast with previous works, where the blanket posteriors are simply assumed to be independent [19, 28], we applied the chain rule to (7), obtaining

$$\Pr(B^{X_0}, \dots, B^{X_{n-1}} \mid D) = \prod_{i=0}^{n-1} \Pr\left(B^{X_i} \mid \left\{B^{X_j}\right\}_{j=0}^{i-1}, D\right). \quad (8)$$

In this way, the posterior probability of each blanket can be described in terms of conditional probabilities, using the training dataset  $D$  as evidence, together with



the blanket of the other variables. Thus, the joint posterior of all the blankets is computed taking advantage of how the blankets are mutually related, instead of assuming them to be independent. The correctness of the proposed method is discussed in Appendix A. Details about how the BJP scoring function proceeds are presented below.

The computation of  $\Pr(B^{X_0}, \dots, B^{X_{n-1}} \mid D)$  has to be done progressively, first calculating the posterior of the blanket of a variable, and then, the knowledge obtained so far can be used as evidence to compute the posterior of the blanket of other variables. However, this decomposition is not unique, since each possible ordering for the variables is associated to a particular decomposition. The basic idea underlying the computation of BJP is to sort the blankets by their size (that is, the degree of the nodes in the graph) in ascending order. This allows a series of inference steps, in order to avoid the computation of expensive and unreliable probabilities, thus improving data efficiency. This is due to the fact that as the size of the blanket increases, greater amounts of data are required for accurately estimating its posterior probability. By using the proposed strategy, the blanket posteriors of variables with fewer neighbors are computed first, and this information is used as evidence when computing the posteriors for variables with bigger blankets. As a result, the information obtained from the more reliable blanket posteriors is used for computing less reliable blankets posteriors.

Now consider an example probability distribution  $\Pr(X_V)$  with four variables  $X = \{X_0, X_1, X_2, X_3\}$ , represented by a MN whose independence structure  $G$  is given by the graph of Figure 1. One possible way of sorting its nodes by their degree in ascending order is represented by the vector  $(X_1, X_2, X_3, X_0)$ , and according to this ordering the blankets joint posterior is decomposed as

$$\begin{aligned} \Pr(B^{X_0}, B^{X_1}, \dots, B^{X_{n-1}} \mid D) &= \Pr(B^{X_1} \mid D) \\ &\quad \times \Pr(B^{X_2} \mid B^{X_1}, D) \\ &\quad \times \Pr(B^{X_3} \mid B^{X_1}, B^{X_2}, D) \\ &\quad \times \Pr(B^{X_0} \mid B^{X_1}, B^{X_2}, B^{X_3}, D). \end{aligned}$$

This example allows us to illustrate the intuition behind BJP, since the sample

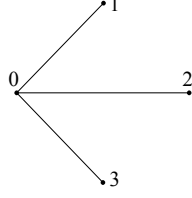


Figure 1: Example of an undirected graph with 4 nodes and hub topology

complexity of the blanket posterior for variables  $X_1$ ,  $X_2$ , and  $X_3$  is lower than that of  $X_0$ . Moreover, in this example it is clear that the posterior distribution of  $B^{X_0}$  is not independent of the posterior distributions of  $B^{X_1}$ ,  $B^{X_2}$  and  $B^{X_3}$ . Clearly, the posterior of  $B^{X_0}$  is harder to evaluate than the posterior of the remaining variables, and then, computing  $\Pr(B^{X_0}|B^{X_1}, B^{X_2}, B^{X_3}, D)$  could be more informative than only computing  $\Pr(B^{X_0}|D)$  independently of the rest of blankets.

Given an undirected graph  $G$ , denote  $\psi$  the ordering vector which contains the variables sorted by their degree in ascending order. Therefore, we reformulate (8) as

$$BJP(G) = \prod_{i=0}^{n-1} \Pr \left( B^{\psi_i} \middle| \left\{ B^{\psi_j} \right\}_{j=0}^{i-1}, D \right). \quad (9)$$

We now proceed to express the posterior of a blanket in terms of probabilities of conditional independence and dependence assertions. The computation of  $\Pr(B^{\psi_i} | \{B^{\psi_j}\}_{j=0}^{i-1}, D)$  can be derived from the posterior of the independences and dependences represented by each blanket:

$$\Pr \left( B^{\psi_i} \middle| \left\{ B^{\psi_j} \right\}_{j=0}^{i-1}, D \right) = \prod_{\psi_k \notin B^{\psi_i}} \Pr \left( \langle \psi_i \perp \psi_k | B^{\psi_i} \rangle \middle| \left\{ B^{\psi_j} \right\}_{j=0}^{i-1}, D \right) \times \prod_{\psi_k \in B^{\psi_i}} \Pr \left( \langle \psi_i \not\perp \psi_k | B^{\psi_i} \setminus \{\psi_k\} \rangle \middle| \left\{ B^{\psi_j} \right\}_{j=0}^{i-1}, D \right). \quad (10)$$

In this way, the whole score is the product of the posterior probability of each blanket, computed in terms of posterior probabilities conditioned in other blankets. The particular way of determining the posterior of each blanket of (10) is inspired by the *Markov blanket closure* [28, Definition 2], which is a set

of independence and dependence assertions formally proven to determine a MN structure.

The two factors in (10) will be interpreted as follows:

- The first product computes the probability of independence between  $\psi_i$  and its non-adjacent variables, conditioned on its blanket, given the previously computed blankets and the dataset  $D$ . It can be computed as

$$\Pr \left( \langle \psi_i \perp \psi_k | B^{\psi_i} \rangle \middle| \left\{ B^{\psi_j} \right\}_{j=0}^{i-1}, D \right) = \begin{cases} \Pr(\langle \psi_i \perp \psi_k | B^{\psi_i} \rangle \mid D) & \text{if } i < k, \\ 1 & \text{if } i > k. \end{cases} \quad (11)$$

Here,  $i < k$  indexes over the variables for which the blanket posterior probability is not already computed. For the remaining variables the posterior of independence will be simply inferred as 1. With this strategy, the score simply uses the information in the evidence, since the independence is determined by the blanket of  $\psi_k$ . The rationale behind this inference is that for cases  $i > k$ , the blanket of  $\psi_k$  has already been computed. As it will be proved in more detail in Appendix A,  $B^{\psi_k}$  already contains information about the independence of  $\psi_i$  and  $\psi_k$ . By considering the local Markov property for the blanket of  $\psi_k$ , and the fact that  $\psi_k$  is not in the blanket of  $\psi_i$ , the opposite must also be true (as these are undirected edges).

- The second product in (10) computes the posterior probability of dependence between  $\psi_i$  and its adjacent variables, conditioned on its remaining neighbors, given the blankets computed previously and the dataset  $D$ . It

can be computed as

$$\Pr \left( \langle \psi_i \not\perp \psi_k | B^{\psi_i} \setminus \{\psi_k\} \rangle \left| \left\{ B^{\psi_j} \right\}_{j=0}^{i-1}, D \right. \right) = \begin{cases} \Pr(\langle \psi_i \not\perp \psi_k | B^{\psi_i} \setminus \{\psi_k\} \rangle \mid D) & \text{if } i < k, \\ 1 & \text{if } i > k. \end{cases} \quad (12)$$

Here, again  $i < k$  indexes over the variables for which the blanket posterior is not already computed. For the remaining variables the posterior of dependence will be inferred as 1. Again, the score use the evidence information, since the independence is determined by the blanket of  $\psi_k$ .

For the sake of clarity, Appendix B shows the complete computation of the BJP score for the graph of Figure 1.

The only approximation in BJP is made in (10), by assuming that all the independence and dependence assertions that determine the blanket of a variable  $\psi_i$  are mutually independent. This is a common assumption, made implicitly by all the constraint-based MN structure learning algorithms [23], and also by the MPL score and the IB-score. For the computation of the posterior probabilities of independence  $\Pr(\langle \psi_i \perp \psi_k | B^{\psi_i} \rangle \mid D)$  and dependence  $\Pr(\langle \psi_i \not\perp \psi_k | B^{\psi_i} \setminus \{\psi_k\} \rangle \mid D)$  used in (11) and (12), respectively, BJP uses the Bayesian test of [38, 34, 39], in the same way as the IB-score explained in the previous section. Precisely, this statistical test computes the posterior of independence and dependence assertions, and has been proven to be statistically consistent in the limit of infinite data.

We now discuss the computational complexity of the score. For a fixed structure, the computational cost is directly determined by the number of statistical tests that it is required to perform on data. Recall that the computational cost of each test is lineal in the number of variables involved and the number of data points [35]. As stated in (9), BJP computes the posterior probability of the blanket for the  $n$  variables of the domain. For each, it is required to perform  $n - 1$  statistical tests on data, by using (10). Then, one half of the tests are in-

ferred when computing the posterior of independences and dependences of (11) and (12). Thus, only  $\frac{n(n-1)}{2}$  tests are required for computing the BJP score of a structure.

We end this section with the optimization proposed in this work for learning the structure with the BJP score. The naïve optimization consists in maximizing over all the possible undirected graphs for some specific problem domain, as in (1), computing with (9) the score for each structure. Since the discrete optimization space of the possible graphs  $\mathcal{G}$  grows rapidly with the number of variables  $n$ , the search is clearly intractable even for small domain sizes. Hence, in this work we test the performance of BJP with brute force only for small domains. For larger domains we use the IBBP-HC algorithm, as an efficient approximate solution proposed in [28].

The optimization made by IBBP-HC is a simple heuristic hill-climbing procedure. The search is initialized by computing the score for an empty structure with no edges, and  $n$  nodes. The hill-climbing search starts with a loop that iterates by selecting the next candidate structure at each iteration. A naïve implementation of hill-climbing would select the neighbor structure with maximum score, computing the score for the  $\binom{n}{2}$  neighbors that differ in one edge. Such expensive computation is avoided by selecting the next candidate with a heuristic that flips the most promising edge (i.e., the edge with lower local contribution to the score). Once the next candidate is selected, its score is computed to be compared to the best scoring structure found so far. The algorithm stops when the neighbor proposed does not improve the current score.

#### 4. Experimental evaluation

This section presents several experiments in order to determine the merits of BJP in practical terms. We compare BJP against two recently proposed scoring functions that approximate the posterior of structures: the Marginal Pseudo Likelihood (MPL) score [19], and the Independence-based score (IB-score) [28]. To the best of our knowledge, there are no other scoring functions

in the literature of MNs for scoring graphical independence structures by using  $P(G \mid D)$ .

Two sets of experiments are presented, one from low-dimensional problems, and another for high-dimensional problems. For the low-dimensional setting, we used brute force (i.e., exhaustive search) to study the convergence of the scoring functions to the exact solution. The goal is to prove experimentally that the sample complexity for successfully learning the exact structure of BJP can be better than for the competitors. For the high-dimensional setting, we used hill-climbing optimization for all the scoring functions. This experiments were performed in order to prove that, by using a similar search strategy, BJP can identify structures with fewer structural errors than the competitor scores. The software to carry out the experiments has been developed in Java, and it is publicly available<sup>1</sup>.

For the experiments we selected a set of networks where the topologies exhibit irregularities, which is a common property in many real-world networks [40]. According to [41], the irregularity of an undirected graph can be computed by summing the imbalance of its edges:

$$irr(G) = \sum_{(i,j) \in E(G)} |d_G(i) - d_G(j)|, \quad (13)$$

where  $d_G(i)$  is the degree of the node  $i$  in that graph. Clearly  $irr(G) = 0$  if and only if  $G$  is regular. For non-regular graphs  $irr(G)$  is a measure of the lack of regularity. Since BJP can infer complex statistical tests from other more simpler tests performed before, we used the irregularity of the underlying structure as an external control variable that determines how important is the independence assumption between blankets for decomposable scores.

#### 4.1. Consistency experiments

A MN scoring function is consistent when the structure which maximizes the score over all the possible structures is the correct one, in the limit of

---

<sup>1</sup><http://dharma.frm.utn.edu.ar/papers/bjp>

infinite data. However, in practice the data is often too scarce to satisfy this condition, and the sample size needed to reach the correct structure varies across different scoring functions. This is referred to as the *sample complexity* of the score. The experiments here presented were carried out in order to measure the sample complexity of the three different scoring functions known to compute the posterior of structures: MPL, IB-score and BJP. This is achieved by measuring their ability to return, by brute force, the exact independence structure of the MN which generated the data.

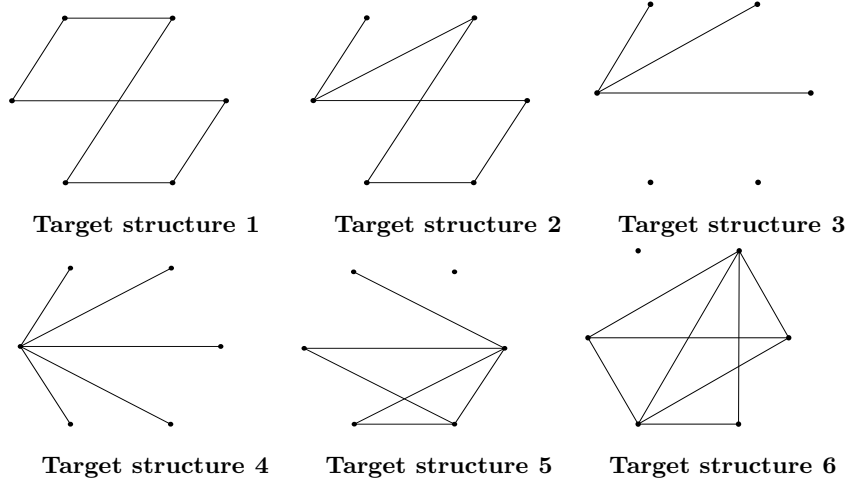


Figure 2: Independence structures for the first set of experiments: model 1 is regular ( $irr = 0$ ); model 2 has  $irr = 10$ ; model 3 has  $irr = 18$ ; model 4 has  $irr = 20$ ; models 5 and 6 have the maximum irregularity for six variables ( $irr = 26$ ).

To make this comparative study, we selected the six different target structures shown in Figure 2. These graphs represent different cases of irregularity, according to (13). The first target structure is regular ( $irr = 0$ ), the second has a little irregularity, the third and fourth structures are irregular structures with a hub topology, and the fifth and sixth target structures have maximum irregularity for  $n = 6$ . As mentioned before, the irregularity is used here as a parameter for determining how important is the independence assumption between blankets for decomposable scores. Thus, in terms of sample complexity, we expect

larger improvements of BJP over the competitors when the irregularity of the underlying structure increases.

For constructing a probability distribution from these independence structures according to (2), random numeric values were assigned to their maximal clique factors, sampled independently from a uniform distribution over  $(0, 1)$ . Ten distributions were generated for each target structure, considering only binary discrete variables. Then, for each one, ten different random seeds were used to obtain 100 datasets for each graph, by using the Gibbs sampling tool of the open-source Libra toolkit [42]. The Gibbs sampler was run with 100 burn-in and 1000 sampling iterations, as commonly used in other works [12, 28, 13].

Since we have  $n = 6$  variables, the search space consists of  $2^{\binom{6}{2}} = 32768$  different undirected graphs. The experiment consisted of evaluating the number of true structures returned by each score over the 100 datasets. This is called here the success rate of the scoring function. The success rate is computed for increasing dataset sizes  $\mathcal{N}_D = \{250, 500, 1000, 2000, 4000, 8000\}$ . Of course, since greater sizes of the dataset lead to better estimations,  $\mathcal{N}_D$  affects the quality of the structure learned. Therefore, a score is considered better than another score when its success rate converges to 1 with lower values of  $\mathcal{N}_D$ .

Table 1 shows the results of the experiment. The first column shows the target structures, the second shows their irregularity, the third shows each sample size  $\mathcal{N}_D$  used, and the fourth shows the success rate. For all the cases, it can be seen how the success rate of the three scoring functions grows with the sample size  $\mathcal{N}_D$ . The results in the fourth column show that BJP has a better success rate in almost all cases. For all the cases, MPL has a slower convergence than IB-score and BJP. This is interesting, since MPL has not been compared before with other approximations of  $\Pr(G|D)$ , and the experimental results shown in [19] only compares the quality obtained by using the score with a local hill-climbing search mechanism against standard constraint-based algorithms. For structures 1 and 2, IB-score shows better convergence than BJP, but they would eventually converge similarly for greater  $\mathcal{N}_D$  sizes. This is an expected result, because these structures are regular, and the approximation of BJP and IB-



	Target structure	$Irr$	$\mathcal{N}_D$	Success rate		
				MPL	IB-score	BJP
<b>1</b>		0	250	0.00	0.00	0.00
			500	0.00	0.00	<b>0.01</b>
			1000	0.01	<b>0.05</b>	0.03
			2000	0.04	<b>0.15</b>	0.12
			4000	0.15	<b>0.25</b>	0.21
			8000	0.28	<b>0.35</b>	0.34
<b>2</b>		10	250	0.00	0.00	0.000
			500	0.00	0.00	<b>0.01</b>
			1000	0.00	<b>0.04</b>	0.02
			2000	0.02	0.15	<b>0.16</b>
			4000	0.10	<b>0.27</b>	0.25
			8000	0.18	0.39	0.39
<b>3</b>		18	250	0.00	<b>0.06</b>	0.04
			500	0.03	0.09	<b>0.12</b>
			1000	0.10	0.17	<b>0.19</b>
			2000	0.17	0.22	<b>0.27</b>
			4000	0.22	0.45	<b>0.49</b>
			8000	0.34	0.58	<b>0.61</b>
<b>4</b>		20	250	0.00	0.00	0.00
			500	0.00	<b>0.03</b>	0.02
			1000	0.00	0.06	<b>0.10</b>
			2000	0.00	0.14	<b>0.18</b>
			4000	0.00	0.29	<b>0.36</b>
			8000	0.00	0.44	<b>0.50</b>
<b>5</b>		26	250	0.00	0.01	0.01
			500	0.00	<b>0.02</b>	0.01
			1000	0.00	0.10	<b>0.11</b>
			2000	0.00	0.23	<b>0.26</b>
			4000	0.03	<b>0.56</b>	0.54
			8000	0.21	0.75	<b>0.76</b>
<b>6</b>		26	250	0.00	0.00	0.00
			500	0.00	0.00	0.00
			1000	0.00	0.04	<b>0.13</b>
			2000	0.00	0.28	<b>0.37</b>
			4000	0.02	<b>0.66</b>	0.61
			8000	0.27	0.80	<b>0.82</b>

Table 1: Success rate of BJP, IB-score and MPL over 100 datasets for the target structures on Figure 2. Rates in bold face correspond to the best case.

score are very similar for computing  $\Pr(G|D)$ . In contrast, for structures 3, 4, 5 and 6, BJP has in general the best success rate. This is also an expected result, according to the irregularity of the underlying structures. Accordingly, the best improvement of BJP over IB-score is for model 6 (which has maximal

irregularity) and  $\mathcal{N}_D = \{1000, 2000\}$ , with an improvement of success rate of up to 9%. When compared with MPL, BJP obtains the best improvement in success rate of up to 59%, also for model 6 and  $\mathcal{N}_D = \{4000\}$ .

In general, these results are consistent with the hypothesis of this work, since BJP has been designed to improve the computation of  $\Pr(G|D)$ , and the irregularity highlights the cases where an improvement of the sample complexity is expected, due to the independence assumption between blankets made by the state-of-the-art scores. The following section shows the performance of the three scoring functions for more complex domains.

#### 4.2. Structural errors analysis

In this section, experiments in the higher-dimensional setting are presented. For this, we evaluate the quality of the structures learned by using an approximate search mechanism. The BJP score and the IB-score were tested with the IBMAP-HC algorithm proposed in [28], briefly explained at the end of Section 3. The MPL scoring function was tested with the most efficient optimization algorithm proposed in [19], described in Section 2.2.

The goal in the experiments is to show how the BJP score can improve the quality of the structures learned over the competitor scores. For this, the selected graphs capture the properties of several real-world problems, where the target structure has few nodes with large degrees, and the remaining nodes have very small degree. Examples of problems with this characteristic include gene networks, protein interaction networks and social networks [40]. Thus, for this comparative study, we used three types of structures: networks with hub topologies, scale-free networks generated by the Barabasi-Albert model [43], and real-world networks, taken from the sparse matrix collection of [44]. These structures have an increasing complexity both in  $n$  and in  $irr$ . The hub networks are shown in Figure 3, the scale-free networks are shown in Figure 4, and the real-world networks are shown in Figure 5.

For each target structure we generated 10 random distributions and 10 random samples for each distribution, with the Gibbs sampler tool of the Libra

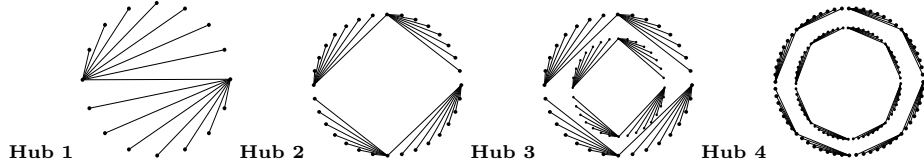


Figure 3: Structures with a hub topology and 16, 32, 64 and 128 nodes

toolkit. Thus, a total of 100 datasets were obtained for each graph, with the same procedure explained in the previous section. As a quality measure, we report the average edge Hamming distance between the hundred learned structures and the underlying one, computed as the sum of false positives and false negatives in the learned structure. As in the previous section, the algorithms were executed for increasing dataset sizes  $\mathcal{N}_D = \{250, 500, 1000, 2000, 4000, 8000\}$ , to assess how their accuracy evolves with data availability.

Table 2 shows the comparison of BJP against MPL and IB-score for the hub structures of Figure 3. The table shows the structures, their sizes  $n$ , and their irregularities, in the first, second and third columns, respectively. The dataset sizes  $\mathcal{N}_D$  are in the fourth column. The fifth column shows the average

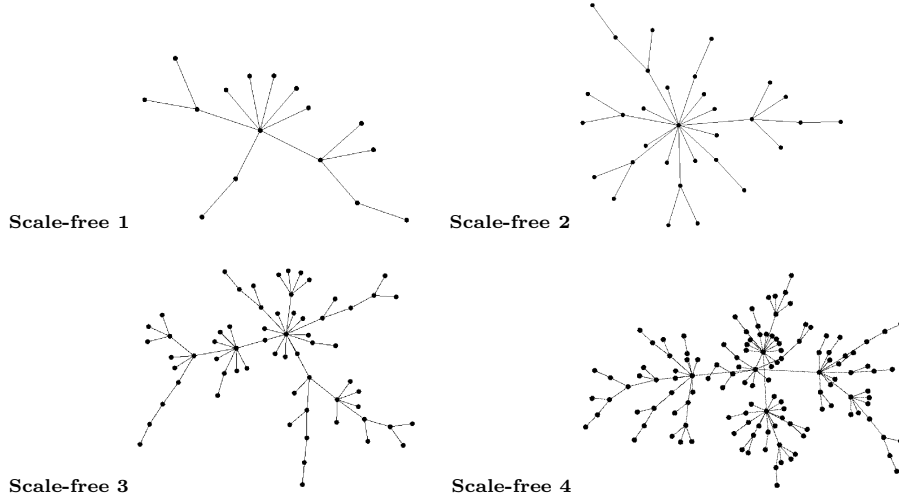


Figure 4: Scale-free structures with 16, 32, 64 and 128 nodes

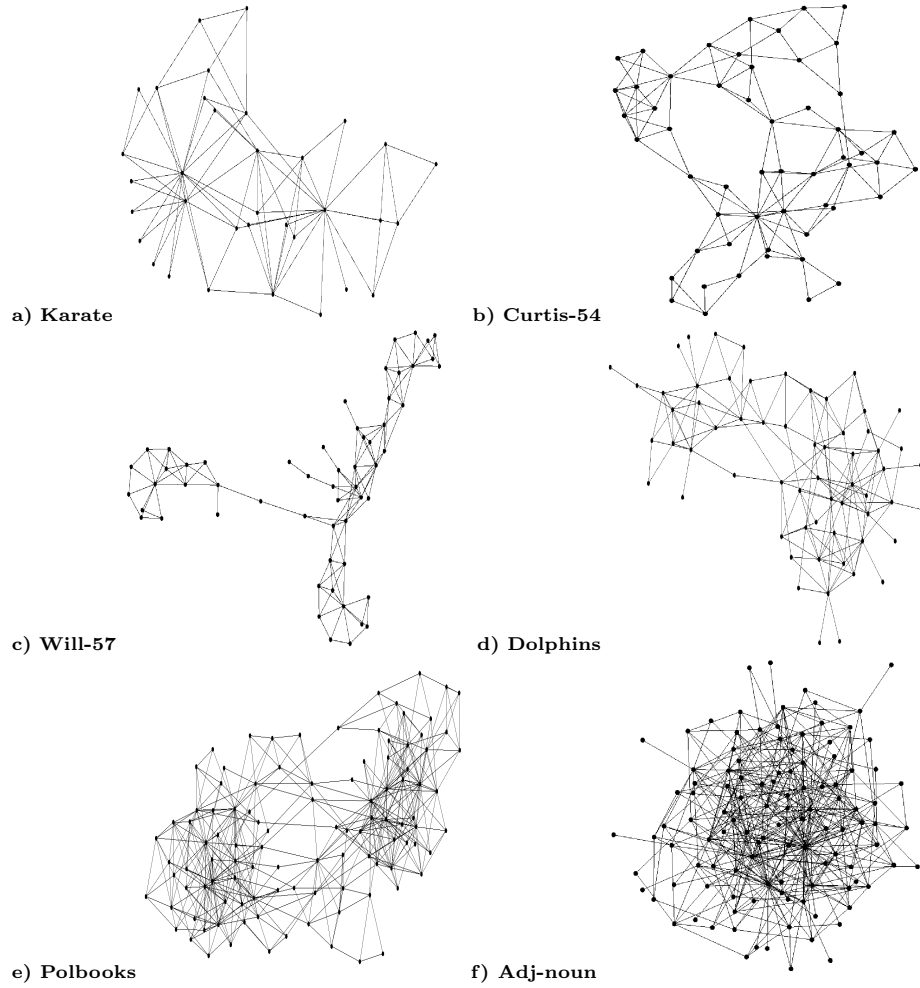


Figure 5: Real-world networks

and standard deviation of the Hamming distance over the 100 repetitions. The sixth column shows the corresponding runtimes (in seconds)<sup>2</sup>. When analyzing these results, it can be seen that for all the algorithms the more complex the underlying structure (determined by  $n$  and  $irr$ ), the larger is the number of structural errors for any score and any value of  $\mathcal{N}_D$ . The results show that BJP

---

<sup>2</sup>All the experiments were performed on an Intel(R) Core(TM) i7-4770 CPU, with 3.40GHz, and 32 GB of main memory.

Target structure	$n$	$irr$	$\mathcal{N}_D$	Hamming distance			Runtime		
				MPL	IB-score	BJP	MPL	IB-score	BJP
Hub 1	16	392	250	13.11 (0.07)	12.36 (0.11)	<b>12.14</b> (0.14)	0.16 (0.00)	0.20 (0.00)	<b>0.06</b> (0.01)
			500	11.76 (0.06)	9.92 (0.09)	<b>9.42</b> (0.11)	0.14 (0.02)	0.29 (0.05)	<b>0.11</b> (0.01)
			1000	10.46 (0.05)	7.80 (0.11)	<b>7.20</b> (0.12)	<b>0.19</b> (0.02)	0.74(0.02)	0.25 (0.04)
			2000	9.40 (0.06)	6.04 (0.11)	<b>5.40</b> (0.11)	<b>0.41</b> (0.06)	2.39 (0.08)	0.60 (0.07)
			4000	8.19 (0.05)	4.06 (0.12)	<b>3.94</b> (0.10)	<b>1.09</b> (0.017)	6.75 (0.22)	1.34 (0.02)
			8000	7.26 (0.05)	3.16 (0.10)	<b>2.88</b> (0.10)	2.908 (0.052)	17.53 (0.59)	<b>2.59</b> (0.02)
Hub 2	32	1916	250	27.22 (0.12)	25.73 (0.09)	<b>25.02</b> (0.12)	0.42 (4.94)	0.81 (0.01)	<b>0.39</b> (0.00)
			500	24.34 (0.11)	22.00 (0.11)	<b>19.98</b> (0.15)	<b>0.59</b> (0.00)	1.50 (0.01)	0.92 (0.01)
			1000	21.53 (0.10)	17.50 (0.12)	<b>15.41</b> (0.16)	<b>1.35</b> (0.02)	3.87 (0.04)	2.15 (0.02)
			2000	18.96 (0.08)	12.86 (0.13)	<b>11.63</b> (0.11)	<b>3.00</b> (0.05)	11.39 (0.14)	5.28 (0.05)
			4000	16.68 (0.08)	9.36 (0.12)	<b>8.36</b> (0.11)	<b>7.67</b> (0.10)	29.32 (0.36)	11.63 (0.09)
			8000	14.56 (0.07)	7.06 (0.10)	<b>6.96</b> (0.10)	<b>22.45</b> (0.28)	76.584 (1.03)	23.75 (0.18)
Hub 3	64	6624	250	60.49 (0.21)	56.55 (0.12)	<b>54.03</b> (0.18)	3.09 (0.03)	1.79 (0.02)	<b>1.37</b> (0.00)
			500	52.92 (0.19)	50.60 (0.14)	<b>44.88</b> (0.20)	4.90 (63.37)	4.96 (0.07)	<b>3.86</b> (0.05)
			1000	46.17 (0.19)	42.33 (0.19)	<b>36.35</b> (0.25)	<b>10.33</b> (0.11)	17.24 (0.22)	10.39 (0.12)
			2000	40.31 (0.18)	33.49 (0.24)	<b>29.21</b> (0.29)	<b>24.73</b> (0.28)	57.95 (0.81)	25.991 (0.38)
			4000	34.97 (0.18)	26.31 (0.25)	<b>22.47</b> (0.30)	<b>61.75</b> (0.66)	180.92 (3.02)	63.64 (0.83)
			8000	30.55 (0.17)	20.87 (0.29)	<b>19.44</b> (0.31)	207.48 (2.08)	627.50 (11.27)	<b>156.24</b> (3.15)
Hub 4	128	24496	250	134.28 (0.35)	120.11 (0.14)	<b>112.43</b> (0.28)	58.92 (0.32)	<b>5.86</b> (0.13)	8.31 (0.13)
			500	113.96 (0.28)	110.03 (0.24)	<b>97.25</b> (0.37)	78.53 (0.49)	26.56 (0.42)	<b>25.14</b> (0.37)
			1000	98.24 (0.29)	95.01 (0.29)	<b>78.39</b> (0.44)	129.33 (0.77)	101.26 (1.05)	<b>74.80</b> (0.77)
			2000	84.27 (0.26)	78.78 (0.34)	<b>61.35</b> (0.54)	259.68 (1.74)	331.32 (3.19)	<b>198.77</b> (2.14)
			4000	72.70 (0.23)	65.17 (0.52)	<b>52.11</b> (0.75)	777.84 (6.36)	1252.88 (19.89)	<b>473.05</b> (6.97)
			8000	62.59 (0.26)	52.95 (0.78)	<b>47.04</b> (1.03)	3102.53 (28.43)	4913.07 (89.81)	<b>1185.91</b> (23.83)

Table 2: Structures with hub topology: average and standard deviation of the Hamming distance and runtime (in seconds) over 100 repetitions. The best average results are in bold.

obtains the best performance for all the cases, reducing the number of average errors of the structures learned by MPL and IB-score. It can be seen that, for all the target structures, again MPL has the slowest convergence in  $\mathcal{N}_D$ . When compared with both MPL and IB-score, the improvements of the BJP score are larger as the complexity ( $n$  and  $irr$ ) grows. These improvements are statistically significant for all the cases against MPL. Against IB-score, the improvements of BJP are statistically significant for all the cases, except three. In general, these results confirm that the approximation of BJP is more accurate as  $n$  and  $irr$  grow. In terms of the respective runtimes, the optimization using the BJP score obtains in general runtimes comparable to MPL and IB-score. For the case of Hub 4, BJP shows the best runtime for all the cases where  $\mathcal{N}_D > 250$ . This is because the more complex the underlying structure the better the convergence of the BJP score to correct structures.

Table 3 shows the comparison of BJP against MPL and IB-score for the scale-free networks of Figure 4. The information of the table is organized in the same

Target structure	$n$	$irr$	$\mathcal{N}_D$	Hamming distance			Runtime		
				MPL	IB-score	BJP	MPL	IB-score	BJP
Scale-free 1	16	364	250	12.35 (0.35)	11.30 (1.63)	<b>11.20</b> (1.23)	0.12 (0.01)	0.33 (0.11)	<b>0.12</b> (0.02)
			500	10.63 (0.26)	10.00 (1.45)	10.00 (1.59)	<b>0.11</b> (0.01)	0.40 (0.10)	0.16 (0.03)
			1000	<b>9.14</b> (0.28)	7.10 (1.64)	7.30 (1.15)	<b>0.25</b> (0.02)	0.76 (0.16)	0.35 (0.03)
			2000	7.53 (0.23)	<b>5.10</b> (1.07)	5.20 (1.04)	<b>0.70</b> (63.75)	1.88 (0.41)	0.71 (0.10)
			4000	6.21 (0.22)	3.70 (0.75)	<b>3.50</b> (0.90)	1.90 (0.14)	3.87 (1.10)	<b>1.41</b> (0.13)
			8000	4.92 (0.22)	2.30 (1.00)	2.30 (1.11)	6.45 (0.71)	7.91 (1.85)	<b>2.91</b> (0.27)
Scale-free 2	32	1612	250	27.51 (0.45)	26.50 (1.74)	<b>25.88</b> (2.00)	<b>0.50</b> (0.03)	0.92 (0.22)	0.56 (0.24)
			500	24.08 (0.46)	22.40 (2.13)	<b>20.38</b> (2.72)	<b>0.78</b> (0.05)	1.34 (0.25)	0.95 (0.24)
			1000	20.82 (0.42)	18.30 (2.00)	<b>17.12</b> (2.15)	2.11 (0.16)	4.34 (1.15)	<b>1.73</b> (0.33)
			2000	18.27 (0.37)	13.60 (1.34)	<b>12.12</b> (1.60)	5.18 (0.35)	19.52 (8.80)	<b>5.20</b> (1.92)
			4000	16.13 (0.31)	<b>10.40</b> (1.77)	10.50 (2.03)	12.57 (0.81)	77.37 (36.80)	<b>10.51</b> (2.97)
			8000	14.41 (0.33)	<b>6.56</b> (1.70)	7.00 (1.28)	41.33 (3.86)	354.14 (207.98)	<b>25.32</b> (10.33)
Scale-free 3	64	6428	250	59.11 (0.91)	57.75 (5.67)	<b>55.33</b> (2.29)	4.73 (0.24)	3.83 (3.35)	<b>2.11</b> (1.10)
			500	50.14 (0.81)	52.00 (7.01)	<b>44.00</b> (8.64)	8.20 (0.45)	9.85 (4.38)	<b>6.06</b> (2.92)
			1000	43.05 (0.73)	43.25 (13.98)	<b>36.00</b> (11.04)	19.54 (1.03)	29.05 (21.09)	<b>13.87</b> (6.11)
			2000	36.71 (0.74)	33.50 (9.97)	<b>27.67</b> (8.26)	46.99 (2.34)	95.44 (49.23)	<b>46.06</b> (9.17)
			4000	31.37 (0.56)	26.25 (4.93)	<b>21.33</b> (2.29)	122.24 (6.49)	275.86 (69.80)	<b>99.06</b> (18.66)
			8000	27.52 (0.57)	19.00 (3.71)	<b>16.00</b> (7.15)	433.09 (22.47)	1124.33 (841.27)	<b>221.92</b> (4.05)
Scale-free 4	128	26188	250	131.42 (1.94)	123.20 (1.09)	<b>116.40</b> (2.73)	72.69 (3.47)	<b>6.71</b> (1.29)	12.50 (4.35)
			500	109.44 (1.75)	110.70 (2.96)	<b>101.00</b> (3.85)	106.37 (5.03)	36.51 (6.21)	<b>30.74</b> (12.14)
			1000	91.47 (1.58)	93.00 (4.02)	<b>83.10</b> (4.75)	196.18 (9.51)	140.10 (17.46)	<b>95.14</b> (31.07)
			2000	77.47 (1.40)	79.20 (5.12)	<b>64.50</b> (5.94)	429.12 (24.21)	403.99 (60.94)	<b>271.96</b> (93.17)
			4000	65.44 (1.30)	62.00 (4.99)	<b>46.50</b> (4.84)	1202.84 (92.93)	1469.52 (283.02)	<b>634.66</b> (95.49)
			8000	57.09 (1.21)	47.90 (3.87)	<b>34.30</b> (3.92)	5103.34 (531.26)	7650.26 (2037.82)	<b>1736.44</b> (437.66)

Table 3: Scale-free networks models: average and standard deviation of the Hamming distance and runtime (in seconds) over 100 repetitions. The best average results are in bold.

way as in Table 2. In contrast with the hub structures, in the scale-free networks the size of the blankets in the underlying network is more variable. This can explain the differences in the trends of the Hamming distance, when compared with the results obtained for the hub networks. It can be seen that for all the cases BJP reduces the number of average errors of MPL. The improvements over MPL are statistically significant for all the cases, except three. When compared with IB-score, BJP shows better average number of errors for all the cases, except four. Those improvements over IB-score are statistically significant only for the Scale-free 4 model. The best improvements of BJP over MPL can be seen for the Scale-free 4 model,  $\mathcal{N}_D = 8000$ , with improvements of more than 20 edges corrected. Against IB-score, the best improvements of BJP can be seen also for the Scale-free 4 model,  $\mathcal{N}_D = 4000$ , with improvements of more than 15 edges corrected. In general, these results confirm that the approximation of BJP is more accurate as  $n$  and  $irr$  grow.

Finally, Table 4 show the results for the real-world networks of Figure 5.

Target structure	$n$	$irr$	$\mathcal{N}_D$	Hamming distance			Runtime		
				MPL	IB-score	BJP	MPL	IB-score	BJP
Karate	34	2044	250	58.60 (2.78)	51.91 (3.74)	<b>51.90</b> (3.59)	5.30 (1.85)	5.01 (1.20)	<b>1.78</b> (0.24)
			500	49.80 (2.26)	44.00 (4.92)	<b>42.20</b> (3.33)	12.01 (4.97)	14.59 (6.37)	<b>2.95</b> (0.29)
			1000	44.00 (2.18)	27.25 (5.25)	<b>26.00</b> (4.55)	22.78 (4.47)	213.57 (75.01)	<b>11.07</b> (3.09)
			2000	40.50 (1.24)	17.12 (4.89)	<b>11.30</b> (3.15)	<b>40.74</b> (3.74)	1220.47 (656.76)	51.54 (10.90)
			4000	38.00 (0.68)	7.88 (2.11)	<b>5.80</b> (1.93)	<b>118.66</b> (12.99)	9557.99 (3025.38)	195.52 (48.87)
			8000	36.60 (0.65)	<b>2.60</b> (0.49)	3.20 (0.82)	<b>320.12</b> (26.48)	30963.00 (3032.01)	665.70 (89.15)
Curtis-54	54	3140	250	76.50 (1.50)	77.00 (2.72)	<b>71.20</b> (2.37)	12.09 (0.51)	11.64 (1.57)	<b>5.42</b> (0.34)
			500	64.40 (1.31)	59.10 (2.33)	<b>56.60</b> (2.00)	28.82 (1.79)	28.49 (3.23)	<b>11.33</b> (0.42)
			1000	52.40 (0.86)	40.10 (1.63)	<b>39.40</b> (2.15)	83.15 (3.25)	83.29 (5.36)	<b>29.48</b> (1.10)
			2000	40.10 (0.82)	22.70 (2.33)	<b>18.90</b> (3.75)	244.77 (9.30)	278.23 (37.50)	<b>86.36</b> (6.45)
			4000	30.30 (1.12)	7.50 (1.55)	<b>4.40</b> (1.47)	689.46 (26.14)	1466.95 (599.15)	<b>240.60</b> (7.56)
			8000	24.00 (0.57)	<b>2.12</b> (0.81)	2.20 (0.64)	2015.04 (54.97)	4665.29 (788.77)	<b>742.01</b> (23.51)
Will-57	57	4156	250	79.50 (1.82)	81.90 (4.17)	<b>79.40</b> (4.25)	13.14 (0.52)	9.67 (1.38)	<b>5.69</b> (0.49)
			500	66.80 (1.34)	63.60 (2.27)	<b>60.70</b> (3.77)	31.49 (1.93)	25.19 (2.42)	<b>12.33</b> (0.92)
			1000	55.60 (1.08)	44.50 (2.21)	<b>42.50</b> (3.78)	85.83 (3.36)	75.41 (6.13)	<b>31.91</b> (2.33)
			2000	47.60 (0.53)	25.30 (2.40)	<b>23.80</b> (2.94)	232.10 (8.80)	245.99 (25.06)	<b>87.38</b> (4.42)
			4000	38.30 (0.89)	10.70 (2.03)	<b>9.40</b> (4.22)	672.76 (19.39)	886.78 (123.50)	<b>274.24</b> (24.75)
			8000	28.90 (0.54)	<b>2.70</b> (0.53)	3.90 (1.75)	2383.00 (72.06)	3077.88 (418.76)	<b>787.45</b> (53.24)
Dolphins	62	6480	250	126.70 (4.00)	126.90 (5.18)	<b>125.20</b> (4.14)	24.02 (2.64)	12.46 (3.02)	<b>7.11</b> (1.35)
			500	106.60 (4.32)	106.10 (6.06)	<b>102.10</b> (5.10)	48.47 (5.52)	30.53 (5.83)	<b>16.73</b> (2.34)
			1000	88.50 (1.90)	71.60 (4.64)	<b>65.90</b> (3.84)	126.57 (12.52)	120.44 (13.49)	<b>55.37</b> (3.75)
			2000	74.20 (2.02)	50.60 (3.41)	<b>47.30</b> (3.52)	349.26 (23.90)	337.56 (26.13)	<b>144.14</b> (12.24)
			4000	63.00 (1.99)	32.50 (2.93)	<b>27.70</b> (3.14)	981.07 (65.15)	1092.66 (102.50)	<b>386.27</b> (25.09)
			8000	50.80 (1.60)	20.60 (1.94)	<b>12.90</b> (2.06)	3591.12 (153.37)	4171.51 (173.19)	<b>1331.72</b> (44.67)
Polbooks	105	30374	250	513.53 (5.94)	435.06 (1.19)	<b>428.53</b> (1.96)	48046.00 (18543.70)	4.33 (0.67)	<b>4.1</b> (0.58)
			500	479.00 (6.30)	428.28 (3.07)	<b>418.27</b> (3.96)	48046.00 (3116.25)	13.48 (2.49)	<b>10.89</b> (2.17)
			1000	439.00 (23.10)	414.11 (4.11)	<b>407.93</b> (4.05)	8532.64 (7682.98)	56.37 (8.57)	<b>29.30</b> (3.81)
			2000	409.43 (7.92)	399.72 (4.38)	<b>393.53</b> (4.68)	1455.56 (203.59)	170.71 (18.60)	<b>85.33</b> (9.02)
			4000	378.86 (7.12)	381.72 (6.04)	<b>375.60</b> (6.41)	2344.57 (722.87)	648.49 (190.16)	<b>246.44</b> (27.90)
			8000	<b>353.14</b> (4.91)	364.07 (7.17)	357.50 (5.30)	4462.01 (1383.34)	3726.04 (1105.15)	<b>971.54</b> (168.34)
Adj-Noun	112	39728	250	505.60 (9.13)	424.30 (0.58)	<b>422.10</b> (2.96)	30664.20 (22276.90)	<b>1.86</b> (0.45)	4.49 (2.12)
			500	461.30 (9.60)	421.20 (1.84)	<b>412.90</b> (2.75)	2146.86 (3246.57)	<b>6.56</b> (2.22)	9.38 (2.32)
			1000	430.90 (6.28)	413.80 (2.35)	<b>401.60</b> (3.26)	521.98 (63.52)	27.30 (5.91)	<b>27.00</b> (5.23)
			2000	399.70 (8.45)	400.80 (3.61)	<b>387.50</b> (5.23)	814.73 (207.03)	108.95 (17.35)	<b>80.53</b> (13.08)
			4000	<b>372.00</b> (4.61)	384.40 (6.23)	373.80 (5.60)	1465.95 (275.36)	449.33 (118.64)	<b>224.86</b> (31.93)
			8000	<b>347.30</b> (6.99)	366.30 (5.90)	356.30 (4.92)	3443.87 (816.50)	2172.69 (749.05)	<b>807.17</b> (243.59)

Table 4: Real networks: average and standard deviation of the Hamming distance and runtime (in seconds) over 100 repetitions. The best average results are in bold.

Again, the information of this table is organized in the same way as in the previous tables. The real network structures are ordered by their complexity (in  $n$  and  $irr$ ). The trends in these results are consistent to those in the previous tables. For all the networks, BJP improves the average quality of the structures learned for all the cases when  $\mathcal{N}_D < 4000$ . When compared with MPL, BJP shows improvements for all the cases, except three. The best improvements of BJP over MPL can be seen for the Polbooks and Adj-noun networks,  $\mathcal{N}_D = 250$ , with improvements of more than 80 edges corrected. This is coherent, since those are the most complex networks, and the best improvements are obtained when data is scarcer. When compared with IB-score, BJP shows improvements for all the cases except three, with no statistically significant differences. The best improvements of BJP over IB-score can be seen for the more complex networks, Polbooks and Adj-noun, with improvements of more than 10 edges corrected. Regarding the runtimes, it can be seen again that BJP tends to improve the runtime over MPL and IB-score for almost all the cases.

In general, the results discussed confirm that BJP always outperforms the competitors when data are scarce. Also, the improvements are greater both in quality and runtime, for the more complex models. This confirms the hypothesis that the approximation proposed by BJP can improve the quality of the learning process.

## 5. Conclusions

In this work we have introduced a novel scoring function for learning the structure of Markov networks. The BJP score computes the posterior probability of independence structures by considering the joint probability distribution of the collection of Markov blankets of the structures. The score computes the posterior of each Markov blanket progressively, using information from other blankets as evidence. The blanket posteriors of variables with fewer neighbors is computed first, and then this information is used as evidence for computing the posteriors for variables with bigger blankets. Thus, BJP can be useful



to improve the data efficiency for problems with complex networks, where the topology exhibits irregularities, such as social and biological networks. In the experiments, BJP scoring proved that can improve the sample complexity of the state-of-the-art competitors. The score is tested by using exhaustive search for low-dimensional problems and by using a heuristic hill-climbing mechanism for higher-dimensional problems. The results show that BJP produces more accurate structures than the state-of-the-art competitors.

We will guide our future work toward the design of more effective optimization methods, since the hill-climbing optimization has two inherent disadvantages: i) by only flipping one edge per step it scales slowly with the number of variables of the domain  $n$ , ii) it is prone to getting stuck in local optima. Moreover, we consider that the properties of BJP score have considerable potential for both further theoretical development, and applications.

## 6. Acknowledgements

This work was supported by Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) [PIP 2013 117], Universidad Nacional del Litoral (UNL) [CAI+D 2011 548] and Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT) [PICT 2014 2627] and [PICT-2012-2731].

# Appendices

## A. - Correctness of BJP

Based on the developments in Section 3, and the analysis in Section 4, we see that the BJP score is a good measure of the fit of the estimated MN to the dataset. In this appendix we are concerned about the correctness of the method used by BJP to compute the posterior of structures. Thus, by correctness we mean that the probability computed by BJP is equivalent to the posterior probability of a MN structure.

In the formulation of the BJP score, the joint distribution of the blankets of  $G$  is calculated by computing the probabilities of conditional independence and dependence assertions contained in the blanket of each variable of the domain. Our discussion in this appendix follows by demonstrating that all the members and non-members of each blanket are unequivocally determined in (10), and therefore, that the joint posterior over these dependences and independences is equivalent to the posterior of the blankets. From [28, Definition 2], the *Markov blanket closure* is a set of independence and dependence assertions that are formally proven to correctly determine a MN structure. This set is obtained by determining the blanket of each variable  $X_i \in X$  with the following set of conditional independence and dependence assertions:

$$\left\{ \langle X_i \perp X_j | B^{X_i} \rangle : X_j \notin B^{X_i} \right\} \cup \left\{ \langle X_i \not\perp X_j | B^{X_i} \setminus \{X_j\} \rangle : X_j \in B^{X_i} \right\}. \quad (\text{A.1})$$

Clearly, this is exactly the same set used by BJP in (10) to compute the posterior of the blanket of each variable of the domain. Since this set determines all members and non-members of a blanket, the posterior of this set of assertions is equivalent to the posterior of the blanket. Then, we demonstrate that such probabilities are correctly estimated by (11) and (12). We proceed by discussing their correctness separately for independence and dependence assertions.

Equation (11) computes the probability of independence between a variable and a non-adjacent variable, conditioned on its blanket, given the previously computed blankets and the dataset  $D$ . In this equation, for the case when  $i < k$ , which indexes over the variables for which the blanket posterior is not already computed, the posterior of the independence assertion  $\langle \psi_i \perp \psi_k | B^{\psi_i} \rangle$  must be computed from data. It is performed by using the Bayesian statistical test of [34], that has been proven to be statistically consistent, since its mean square error tends to 0 as the dataset size tends to infinity. For the case when  $i > k$ , which indexes over the variables for which the blanket posterior is already computed, the independence assertion is inferred as 1, since its independence is determined by the blanket of  $\psi_k$ , which is in the evidence  $\{B^{\psi_j}\}_{j=0}^{i-1}$ . By

definition in (10), this case applies to all the variables  $\psi_k \notin B^{\psi_i}$  (i.e., all the variables that are not connected to  $\psi_i$ ). We argue the correctness for this inference by considering an intuitive equivalence commonly used by constraint-based approaches to perform independence tests that involve smaller number of variables [3, p. 980]. If two variables  $X_i$  and  $X_k$  are not neighbors in  $G$ , then by applying the local Markov property of (3) once for each, we have that  $\langle X_i \perp X_k | B^{X_i} \rangle$  and  $\langle X_i \perp X_k | B^{X_k} \rangle$  hold. Therefore, the inference made is correct.

A similar argument can be given for the case of the dependence assertions. Equation (12) computes the probability of dependence between a variable and an adjacent variable conditioned on its remaining neighbors, given the previously computed blankets and the dataset  $D$ . Again, for the case when  $i < k$ , which indexes over the variables for which the blanket posterior is not already computed, the posterior of the dependence assertion must be computed from data. For the case when  $i > k$ , which indexes over the variables for which the blanket posterior is already computed, the dependence assertion is inferred as 1, since its dependence is determined by the blanket of  $\psi_k$ , which is again in the evidence  $\{B^{\psi_j}\}_{j=0}^{i-1}$ . By definition in (10), this case applies to all the variables  $\psi_k \in B^{\psi_i}$  (i.e., all the variables that are connected to  $\psi_i$ ). Clearly, if two variables  $X_i$  and  $X_k$  are neighbors in  $G$ , there are no sets separating them in the graph. Therefore, the dependence assertion inferred is true.

## B. Example of BJP score computation

This appendix shows a complete example of the computation of the BJP score for the graph of Figure 1. Consider this graph as the independence structure of a probability distribution  $\Pr(V)$ , with  $n = 4$  variables  $V = \{X_0, X_1, X_2, X_3\}$ , represented by a MN. Given a dataset  $D$ , the BJP score can be computed by following the next steps:

- a) Build the vector  $\psi$ , with the nodes sorted by their degree in ascending order:  $\psi = (X_1, X_2, X_3, X_0)$ .

b) By following (9), the computation of  $BJP(G)$  is given by:

$$\begin{aligned} BJP(G) = & \Pr\left(B^{X_1} \middle| D\right) \\ & \times \Pr\left(B^{X_2} \middle| B^{X_1}, D\right) \\ & \times \Pr\left(B^{X_3} \middle| B^{X_1}, B^{X_2}, D\right) \\ & \times \Pr\left(B^{X_0} \middle| B^{X_1}, B^{X_2}, B^{X_3}, D\right). \end{aligned}$$

c) Compute each term of the above expression by following (10), resulting in:

$$\begin{aligned} \Pr\left(B^{X_1} \middle| D\right) &= \Pr\left(\langle X_1 \perp X_2 | X_0 \rangle \middle| D\right) \\ &\quad \times \Pr\left(\langle X_1 \perp X_3 | X_0 \rangle \middle| D\right) \\ &\quad \times \Pr\left(\langle X_1 \not\perp X_0 | \emptyset \rangle \middle| D\right). \\ \Pr\left(B^{X_2} \middle| B^{X_1}, D\right) &= \Pr\left(\langle X_2 \perp X_1 | X_0 \rangle \middle| B^{X_1}, D\right) \\ &\quad \times \Pr\left(\langle X_2 \perp X_3 | X_0 \rangle \middle| B^{X_1}, D\right) \\ &\quad \times \Pr\left(\langle X_2 \not\perp X_0 | \emptyset \rangle \middle| B^{X_1}, D\right). \\ \Pr\left(B^{X_3} \middle| B^{X_1}, B^{X_2}, D\right) &= \Pr\left(\langle X_3 \perp X_1 | X_0 \rangle \middle| B^{X_1}, B^{X_2}, D\right) \\ &\quad \times \Pr\left(\langle X_3 \perp X_2 | X_0 \rangle \middle| B^{X_1}, B^{X_2}, D\right) \\ &\quad \times \Pr\left(\langle X_3 \not\perp X_0 | \emptyset \rangle \middle| B^{X_1}, B^{X_2}, D\right). \\ \Pr\left(B^{X_0} \middle| B^{X_1}, B^{X_2}, B^{X_3}, D\right) &= \Pr\left(\langle X_0 \not\perp X_1 | X_2, X_3 \rangle \middle| B^{X_1}, B^{X_2}, B^{X_3}, D\right) \\ &\quad \times \Pr\left(\langle X_0 \not\perp X_2 | X_1, X_3 \rangle \middle| B^{X_1}, B^{X_2}, B^{X_3}, D\right) \\ &\quad \times \Pr\left(\langle X_0 \not\perp X_3 | X_1, X_2 \rangle \middle| B^{X_1}, B^{X_2}, B^{X_3}, D\right). \end{aligned}$$

d) By replacing Equations (11) and (12) in the factors of the above expression, one half of the tests can be inferred, and only the following probabilities

must be computed from data by using the Bayesian statistical test:

$$\begin{aligned}
\Pr\left(B^{X_1}\middle|D\right) &= \Pr\left(\langle X_1 \perp X_2 | X_0 \rangle \middle| D\right) \times \Pr\left(\langle X_1 \perp X_3 | X_0 \rangle \middle| D\right) \times \Pr\left(\langle X_1 \not\perp X_0 | \emptyset \rangle \middle| D\right). \\
\Pr\left(B^{X_2}\middle|B^{X_1}, D\right) &= 1 \times \Pr\left(\langle X_2 \perp X_3 | X_0 \rangle \middle| D\right) \times \Pr\left(\langle X_2 \not\perp X_0 | \emptyset \rangle \middle| D\right). \\
\Pr\left(B^{X_3}\middle|B^{X_1}, B^{X_2}, D\right) &= 1 \times 1 \times \Pr\left(\langle X_3 \not\perp X_0 | \emptyset \rangle \middle| D\right). \\
\Pr\left(B^{X_0}\middle|B^{X_1}, B^{X_2}, B^{X_3}, D\right) &= 1 \times 1 \times 1.
\end{aligned}$$

The inferred tests are the 1s at each equation.

## References

- [1] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Publishers, Inc., 1988.
- [2] S. L. Lauritzen, Lectures in contingency tables, 2nd Edition, University of Aalborg Press, Aalborg, Denmark, 1982.
- [3] D. Koller, N. Friedman, Probabilistic Graphical Models: Principles and Techniques, MIT Press, 2009.
- [4] S. Li, Markov random field modeling in image analysis, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.
- [5] W. Hwang, J. Kim, Markov network-based unified classifier for face recognition, IEEE Transactions on Image Processing 24 (11) (2015) 4263–4275.
- [6] F. Peng, J. Lu, Y. Wang, R. Yi-Da Xu, C. Ma, J. Yang, N-dimensional markov random field prior for cold-start recommendation, Neurocomputing 191 (2016) 187–199.
- [7] Y. Li, S. A. Pearl, S. A. Jackson, Gene networks in plant biology: approaches in reconstruction and analysis, Trends in plant science 20 (10) (2015) 664–675.

- [8] M. Schmidt, K. Murphy, G. Fung, R. Rosales, Structure learning in random fields for heart motion abnormality detection, in: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, 2008, pp. 1–8. doi:10.1109/CVPR.2008.4587367.
- [9] Y.-W. Wan, G. I. Allen, Y. Baker, E. Yang, P. Ravikumar, Z. Liu, M. Y.-W. Wan, Package xmrf.
- [10] P. Larrañaga, J. Lozano, Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation, Kluwer Pubs, 2002.
- [11] S. Shaky, R. Santana, J. Lozano, A markovianity based optimisation algorithm, Genetic Programming and Evolvable Machines 13 (2) (2012) 159–195.
- [12] D. Lowd, J. Davis, Improving markov network structure learning using decision trees, Journal of Machine Learning Research 15 (2014) 501–532.
- [13] J. Van Haaren, J. Davis, Markov network structure learning: A randomized feature generation approach, in: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, 2012.
- [14] J. Davis, P. Domingos, Bottom-up learning of Markov network structure, in: Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 271–278.
- [15] S. Lee, V. Ganapathi, D. Koller, Efficient structure learning of Markov networks using L1-regularization, in: NIPS, 2006.
- [16] J. Van Haaren, J. Davis, M. Lappenschaar, A. Hommersom, Exploring disease interactions using markov networks, in: Workshops at the Twenty-Seventh AAAI Conference on Artificial Intelligence, 2013.
- [17] G. Claeskens, E. Pircalabelu, L. Waldorp, Constructing graphical models via the focused information criterion, in: Modeling and Stochastic Learning for Forecasting in High Dimensions, Springer, 2015, pp. 55–78.

- [18] H. Nyman, J. Pensar, T. Koski, J. Corander, Context-specific independence in graphical log-linear models, *Computational Statistics* (2014) 1–20.
- [19] J. Pensar, H. Nyman, J. Niiranen, J. Corander, et al., Marginal pseudo-likelihood learning of discrete Markov network structures, *Bayesian analysis*.
- [20] P. Spirtes, C. Glymour, R. Scheines, *Causation, Prediction, and Search*, Adaptive Computation and Machine Learning Series, MIT Press, 2000.
- [21] F. Bromberg, D. Margaritis, V. Honavar, Efficient Markov network structure discovery using independence tests, *JAIR* 35 (2009) 449–485.
- [22] C. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, X. Koutsoukos, Local Causal and Markov Blanket Induction for Causal Discovery and Feature Selection for Classification Part I: Algorithms and Empirical Evaluation, *JMLR* 11 (2010) 171–234.
- [23] F. Schlüter, A survey on independence-based Markov networks learning, *Artificial Intelligence Review* (2012) 1–25.
- [24] S. Della Pietra, V. Della Pietra, J. Lafferty, Inducing Features of Random Fields, *IEEE Trans. PAMI.* 19 (4) (1997) 380–393.
- [25] A. McCallum, Efficiently inducing features of conditional random fields, in: *Proceedings of Uncertainty in Artificial Intelligence (UAI)*, 2003.
- [26] V. Ganapathi, D. Vickrey, J. Duchi, D. Koller, Constrained Approximate Maximum Entropy Learning of Markov Random Fields, in: *Uncertainty in Artificial Intelligence*, 2008, pp. 196–203.
- [27] F. Barahona, On the computational complexity of Ising spin glass models, *Journal of Physics A: Mathematical and General* 15 (10) (1982) 3241–3253.
- [28] F. Schlüter, F. Bromberg, A. Edera, The IBCMAP approach for Markov network structure learning, *Annals of Mathematics and Artificial Intelligence* (2014) 1–27doi:10.1007/s10472-014-9419-5.

- [29] M. Frydenberg, S. L. Lauritzen, Decomposition of maximum likelihood in mixed graphical interaction models, *Biometrika* (1989) 539–555.
- [30] A. P. Dawid, S. L. Lauritzen, Hyper Markov laws in the statistical analysis of decomposable graphical models, *The Annals of Statistics* (1993) 1272–1317.
- [31] J. Hammersley, P. Clifford, Markov fields on finite graphs and lattices.
- [32] T. Cover, J. Thomas, *Elements of information theory*, Wiley-Interscience, New York, NY, USA, 1991.
- [33] A. Agresti, *Categorical Data Analysis*, 2nd Edition, Wiley, 2002.
- [34] D. Margaritis, Distribution-Free Learning of Bayesian Network Structure in Continuous Domains, in: *Proceedings of AAAI*, 2005.
- [35] W. Cochran, Some methods of strengthening the common  $\chi$  tests, *Biometrics*. (1954) 10:417451.
- [36] J. E. Besag, Nearest-neighbour systems and the auto-logistic model for binary data, *Journal of the Royal Statistical Society. Series B (Methodological)* (1972) 75–83.
- [37] I. Tsamardinos, C. Aliferis, A. Statnikov, Algorithms for large scale Markov blanket discovery, in: *FLAIRS*, 2003.
- [38] D. Margaritis, F. Bromberg, Efficient Markov Network Discovery Using Particle Filter, *Comp. Intel.* 25 (4) (2009) 367–394.
- [39] D. Margaritis, S. Thrun, Bayesian network induction via local neighborhoods, in: *Proceedings of NIPS*, 2000.
- [40] T. Silva, L. Zhao, *Machine Learning in Complex Networks*, Springer International Publishing, 2016.  
URL <https://books.google.com.ar/books?id=WdDurQEACAAJ>



- [41] M. O. Albertson, The irregularity of a graph, *Ars Combinatoria* 46 (1997) 219–225.
- [42] D. Lowd, A. Rooshenas, The libra toolkit for probabilistic models, arXiv preprint arXiv:1504.00110.
- [43] A. Barabasi, E. Bonabeau, Scale-free networks, *Scientific American*.
- [44] T. A. Davis, Y. Hu, The university of florida sparse matrix collection, *ACM Transactions on Mathematical Software (TOMS)* 38 (1) (2011) 1.